

18 MAR 2004

WIPO PCT

PCT/JP03/15838

日 本 国 特 許 庁
JAPAN PATENT OFFICE

11.12.03

別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office.

出 願 年 月 日
Date of Application: 2003年 1月 9日

出 願 番 号
Application Number: 特願2003-003038
[ST. 10/C]: [JP2003-003038]

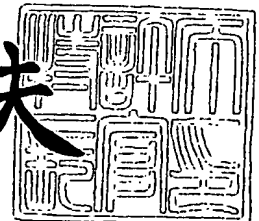
出 願 人
Applicant(s): 独立行政法人 科学技術振興機構
学校法人慶應義塾

PRIORITY DOCUMENT
SUBMITTED OR TRANSMITTED IN
COMPLIANCE WITH
RULE 17.1(a) OR (b)

2004年 3月 4日

特許庁長官
Commissioner,
Japan Patent Office

今 井 康 夫



出証番号 出証特2003-3112562

【書類名】 特許願

【整理番号】 P0656JP

【提出日】 平成15年 1月 9日

【あて先】 特許庁長官殿

【国際特許分類】 G06F 09/00

【発明者】

 【住所又は居所】 横浜市神奈川区六角橋 2 - 2 1 - 4

 【氏名】 山崎 信行

【特許出願人】

 【識別番号】 396020800

 【氏名又は名称】 科学技術振興事業団

【特許出願人】

 【識別番号】 899000079

 【氏名又は名称】 学校法人 慶應義塾

【代理人】

 【識別番号】 100107010

 【弁理士】

 【氏名又は名称】 橋爪 健

【手数料の表示】

 【予納台帳番号】 054885

 【納付金額】 21,000円

【提出物件の目録】

 【物件名】 明細書 1

 【物件名】 図面 1

 【物件名】 要約書 1

 【包括委任状番号】 0013428

【プルーフの要否】 要

【書類名】 明細書

【発明の名称】 コンテキスト切り替え方法及び装置、中央演算装置、コンテキスト切り替えプログラム及びそれを記憶したコンピュータ読み取り可能な記憶媒体

【特許請求の範囲】

【請求項 1】

複数のコンテキストを切り替えるコンテキスト切り替え装置であって、
演算論理ユニット又はメモリアクセスユニットで実行すべきスレッドに関するコンテキストが記憶されたレジスタファイルと、

前記レジスタファイルに接続され、コンテキストをキャッシュするためのコンテキストキャッシュと、

前記レジスタファイルと前記コンテキストキャッシュを接続するコンテキストスイッチ専用バスと、

前記コンテキストキャッシュに記憶されているスレッドのコンテキストを識別するためのスレッド識別子（スレッドID）を記憶するスレッドIDテーブルを有し、演算論理ユニット及びメモリアクセスユニットと並列に接続され、前記コンテキストキャッシュと前記レジスタファイルとの間のデータ伝送を制御するスレッド制御ユニットと

を備え、

前記スレッド制御ユニットは、

コンテキストの切り替えが発生した場合、入力された切り替え命令と新たに入れ替えるスレッド識別子（スレッドID）に基づき前記スレッドIDテーブルを検索し、

新たに入れ替えるコンテキストが記憶されている前記コンテキストキャッシュのアドレスと、実行中のコンテキストが記憶されている前記レジスタファイルのレジスタ識別子を求め、

求めたアドレスに基づき前記コンテキストキャッシュをアクセスし、且つ、求めたレジスタ識別子に基づき前記レジスタファイルをアクセスし、前記レジスタファイル及び／又は前記コンテキストキャッシュのコンテキストを前記コンテキ

ストスイッチ専用バスを介して入れ替え、退避又は復帰を実行する
前記コンテキスト切り替え装置。

【請求項 2】

前記コンテキストスイッチ専用バスは、バス幅をレジスタ長以上とすることで、1度に複数のコンテキストのデータを同時にスワップ、バックアップ又はリストアする請求項 1 に記載のコンテキスト切り替え装置。

【請求項 3】

前記コンテキストキャッシュは、リードポートとライトポートを備え、

前記レジスタファイルは、リードポートとライトポート、コンテキストスイッチ用リードポートとライトポートを備え、

前記コンテキストキャッシュのリードポートとライトポートは、それぞれ前記コンテキストスイッチ専用バスにより前記レジスタファイルのコンテキストスイッチ用ライトポートとリードポートと接続されることを特徴とする請求項 1 に記載のコンテキスト切り替え装置。

【請求項 4】

前記スレッド制御ユニットは、コンテキストキャッシュにキャッシュされた所定数のコンテキストを識別する所定数のスレッド ID テーブルを含むことを特徴とする請求項 1 に記載のコンテキスト切り替え装置。

【請求項 5】

前記スレッド制御ユニットは、コンテキストを入れ替えるスワップ命令を実行する場合、オペレーティングシステム等のソフトウェアからスレッド識別子（スレッド ID）をオペランドとして含むスワップ命令が発行されると、前記レジスタファイルから前記コンテキストキャッシュへ実行中のスレッドのコンテキストを退避すると同時に、前記コンテキストキャッシュから前記レジスタファイルへ新しく入れ替えるスレッドのコンテキストを送り、自動的に必要な数だけ前記レジスタファイル内のデータと前記コンテキストキャッシュ内のデータを入れ替えることを特徴とする請求項 1 乃至 4 のいずれかに記載のコンテキスト切り替え装置。

【請求項 6】

前記スレッド制御ユニットは、コンテキストを退避するバックアップ命令を実行する場合、オペレーティングシステム等のソフトウェアからスレッド識別子（スレッドID）をオペランドとして含むバックアップ命令が発行されると、前記レジスタファイルから前記コンテキストキャッシュへのコンテキストのデータ転送を行い、前記コンテキストキャッシュから前記レジスタファイルへのコンテキストのデータ転送は行わないことを特徴とする請求項1乃至4のいずれかに記載のコンテキスト切り替え装置。

【請求項7】

前記スレッド制御ユニットは、コンテキストを復帰するリストア命令を処理する場合、オペレーティングシステム等のソフトウェアからスレッド識別子（スレッドID）をオペランドとして含むリストア命令が発行されると、前記コンテキストキャッシュから前記レジスタファイルへのコンテキストのデータ転送を行い、前記レジスタファイルから前記コンテキストキャッシュへのコンテキストのデータ転送は行わないことを特徴とする請求項1乃至4のいずれかに記載のコンテキスト切り替え装置。

【請求項8】

前記請求項1乃至7のいずれかに記載のコンテキスト切り替え装置と、
命令及びデータをそれぞれキャッシュする命令キャッシュ及びデータキャッシュと、

前記命令キャッシュから命令をフェッチ及びデコードする命令フェッチユニットと、

レジスタファイルに記憶された命令に従い各種演算を行い、演算結果を前記レジスタファイルに書き戻す演算論理ユニットと、

前記レジスタファイルからオペランドと命令が送られ、前記データキャッシュをアクセスし、ロード又はストアを実行するメモリアクセスユニットと、

前記レジスタファイル、前記演算論理ユニット、前記メモリアクセスユニット及び前記スレッド制御ユニットを並列に接続する演算バスと、
を備えた中央演算装置。

【請求項9】

前記メモリアクセスユニットは、ストア命令の場合はアドレスとデータを前記データキャッシュに送り、データを前記データキャッシュに格納し、ロード命令の場合はアドレスを前記データキャッシュに送り、データを前記データキャッシュから読み出し、読み出したデータを前記レジスタファイルに書き戻す請求項 8 に記載の中央演算装置。

【請求項 10】

演算論理ユニット又はメモリアクセスユニットで実行すべきスレッドに関するコンテキストが記憶されたレジスタファイルと、

前記レジスタファイルに接続され、コンテキストをキャッシュするためのコンテキストキャッシュと、

前記レジスタファイルと前記コンテキストキャッシュを接続するコンテキストスイッチ専用バスと

を備えたコンテキスト切り替え装置を用い、複数のコンテキストを切り替えるコンテキスト切り替え方法であって、

コンテキストの切り替えが発生した場合、入力された切り替え命令と新たに入れ替えるスレッド識別子（スレッド ID）に基付き、前記コンテキストキャッシュに記憶されているスレッドのコンテキストを識別するためのスレッド識別子（スレッド ID）を記憶する前記スレッド ID テーブルを検索し、

新たに入れ替えるコンテキストが記憶されている前記コンテキストキャッシュのアドレスと、実行中のコンテキストが記憶されている前記レジスタファイルのレジスタ識別子を求め、

求めたアドレスに基付き前記コンテキストキャッシュをアクセスし、且つ、求めたレジスタ識別子に基付き前記レジスタファイルをアクセスし、前記レジスタファイル及び／又は前記コンテキストキャッシュのコンテキストを前記コンテキストスイッチ専用バスを介して入れ替え、退避又は復帰を実行する前記コンテキスト切り替え方法。

【請求項 11】

コンテキストを入れ替えるスワップ命令を実行する場合、オペレーティングシステム等のソフトウェアからスレッド識別子（スレッド ID）をオペランドとし

で含むスワップ命令が発行されると、前記レジスタファイルから前記コンテキストキャッシュへ実行中のスレッドのコンテキストを退避すると同時に、前記コンテキストキャッシュから前記レジスタファイルへ新しく入れ替えるスレッドのコンテキストを送り、自動的に必要な数だけ前記レジスタファイル内のデータと前記コンテキストキャッシュ内のデータを入れ替えることを特徴とする請求項10に記載のコンテキスト切り替え方法。

【請求項12】

コンテキストを退避するバックアップ命令を実行する場合、オペレーティングシステム等のソフトウェアからスレッド識別子（スレッドID）をオペランドとして含むバックアップ命令が発行されると、前記レジスタファイルから前記コンテキストキャッシュへのコンテキストのデータ転送を行い、前記コンテキストキャッシュから前記レジスタファイルへのコンテキストのデータ転送は行わないことを特徴とする請求項10に記載のコンテキスト切り替え方法。

【請求項13】

コンテキストを復帰するリストア命令を処理する場合、オペレーティングシステム等のソフトウェアからスレッド識別子（スレッドID）をオペランドとして含むリストア命令が発行されると、前記コンテキストキャッシュから前記レジスタファイルへのコンテキストのデータ転送を行い、前記レジスタファイルから前記コンテキストキャッシュへのコンテキストのデータ転送は行わないことを特徴とする請求項10に記載のコンテキスト切り替え方法。

【請求項14】

演算論理ユニット又はメモリアクセスユニットで実行すべきスレッドに関するコンテキストが記憶されたレジスタファイルと、

前記レジスタファイルに接続され、コンテキストをキャッシュするためのコンテキストキャッシュと、

前記レジスタファイルと前記コンテキストキャッシュを接続するコンテキスト専用バスと

を備えたコンテキスト切り替え装置を用い、コンピュータが複数のコンテキストを切り替えるコンテキスト切り替えプログラムであって、

コンテキストの切り替えが発生した場合、入力された切り替え命令と新たに入れ替えるスレッド識別子（スレッドID）に基付き、前記コンテキストキャッシュに格納されているスレッドのコンテキストを識別するためのスレッド識別子（スレッドID）を記憶する前記スレッドIDテーブルを検索するステップと、

新たに入れ替えるコンテキストが記憶されている前記コンテキストキャッシュのアドレスと、実行中のコンテキストが記憶されている前記レジスタファイルのレジスタ識別子を求めるステップと、

求めたアドレスに基付き前記コンテキストキャッシュをアクセスし、且つ、求めたレジスタ識別子に基付き前記レジスタファイルをアクセスし、前記レジスタファイル及び／又は前記コンテキストキャッシュのコンテキストを前記コンテキストスイッチ専用バスを介して入れ替え、退避又は復帰を実行するステップとをコンピュータに実行させるためのコンテキスト切り替えプログラム。

【請求項15】

演算論理ユニット又はメモリアクセスユニットで実行すべきスレッドに関するコンテキストが記憶されたレジスタファイルと、

前記レジスタファイルに接続され、コンテキストをキャッシュするためのコンテキストキャッシュと、

前記レジスタファイルと前記コンテキストキャッシュを接続するコンテキストスイッチ専用バスと

を備えたコンテキスト切り替え装置を用い、コンピュータが複数のコンテキストを切り替えるコンテキスト切り替えプログラムを記録したコンピュータ読み取り可能な記録媒体であって、

コンテキストの切り替えが発生した場合、入力された切り替え命令と新たに入れ替えるスレッド識別子（スレッドID）に基付き、前記コンテキストキャッシュに格納されているスレッドのコンテキストを識別するためのスレッド識別子（スレッドID）を記憶する前記スレッドIDテーブルを検索するステップと、

新たに入れ替えるコンテキストが記憶されている前記コンテキストキャッシュのアドレスと、実行中のコンテキストが記憶されている前記レジスタファイルのレジスタ識別子を求めるステップと、

求めたアドレスに基付き前記コンテキストキャッシュをアクセスし、且つ、求めたレジスタ識別子に基付き前記レジスタファイルをアクセスし、前記レジスタファイル及び／又は前記コンテキストキャッシュのコンテキストを前記コンテキストスイッチ専用バスを介して入れ替え、退避又は復帰を実行するステップとをコンピュータに実行させるためのコンテキスト切り替えプログラムを記録したコンピュータ読み取り可能な記録媒体。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】

本発明は、コンテキスト切り替え方法及び装置、中央演算装置、コンテキスト切り替えプログラム及びそれを記憶したコンピュータ読み取り可能な記憶媒体に係り、特に、リアルタイム・オペレーティングシステム（RTOS）等のOSにおいてコンテキストの切り替えによるオーバーヘッドの削減を可能にするコンテキスト切り替え方法及び装置、中央演算装置、コンテキスト切り替えプログラム及びそれを記憶したコンピュータ読み取り可能な記憶媒体に関する。ここで、コンテキストとは、例えば、汎用レジスタ、浮動小数点レジスタ、プログラムカウンタ、ステータスレジスタ等、記憶部（例、レジスタファイル）に記憶されている各スレッドの実行のための情報又は現在実行中の状態のことをいう。

【0002】

【従来の技術】

図8に、コンテキスト切り替え動作の説明図を示す。

この図では、複数のコンテキスト（スレッド）が1つの中央演算装置で切り替えられて実行されている例を示している。コンテキストが切り替わる場合、今まで実行していたコンテキストの状態（汎用レジスタ、浮動小数点レジスタ、プログラムカウンタ、ステータスレジスタなどであり、以下、単にコンテキストという場合がある。）を保存し、新しく実行するコンテキストの状態を読み出す必要がある。切り替え動作の際に要する時間をオーバーヘッドといい、オーバーヘッドは、コンテキストが切り替わる度に発生する。

【0003】

従来、コンテキスト切り替え時間を短縮するための技術としては、以下の文献が挙げられる。

特許文献1には、リアルタイムオペレーティングシステムを用いたマルチタスク処理装置において、各タスクに対応して占有される複数のレジスタバンクを設け、コンテキスト等の退避／復帰がレジスタバンクを切り替えることにより行い、デイスパッチ時間を短縮する技術が記載されている。

【0004】

また、特許文献2には、マルチタスク処理を行うマイクロプロセッサにおいて、複数のレジスタに対応してその内容の変化又は非変化を示すビットを設け、タスク切り替えが発生したときに、そのビットに従い、レジスタの内容が変化したときに退避命令を実行し、変化しないとき退避命令を実行しないようにすることで、OSのオーバーヘッドを減少させるレジスタ退避及び復元システムが記載されている。

【特許文献1】

特開平07-141208号

【特許文献2】

特開平09-212371号

【0005】

【発明が解決しようとする課題】

従来の方法では、コンテキストを切り替える場合、中央演算装置内に保持されているコンテキストの状態は、OS等のソフトウェアによりストア命令を用いて1つずつ中央演算装置外の記憶装置に保存される。その後、OS等のソフトウェアは、ロード命令を用いて新しいコンテキストを記憶装置から読み込む。つまり、コンテキストを切り替える度に、コンテキストの保存と読み込みのためのメモリアクセスが数百サイクルから千数百サイクルも生じることになり、大きなオーバーヘッドとなっている。また、従来の方法では、ロード命令やストア命令を用いたソフトウェアによるコンテキストの保存と読み込みを行っているため、1度に1つのデータしか扱うことができない。そのため保存すべき状態が増えるとコンテキストスイッチにかかる時間も増加する。

【0006】

本発明は、以上の点に鑑み、特に、リアルタイムOSなどコンテキストの切り替えが頻繁に発生するアプリケーションにおいて、コンテキストの切り替え（コンテキストスイッチ）によるオーバーヘッドを大幅に削減することを目的とする。本発明は、例えば、コンテキストを切り替える度に、コンテキストの保存と読み込みのためのメモリアクセスが、1サイクルから数サイクルで可能とするコンテキスト切り替え方法及び装置、中央演算装置、コンテキスト切り替えプログラム及びそれを記憶したコンピュータ読み取り可能な記憶媒体を提供することを目的とする。

【0007】

また、本発明は、特に実時間処理システムのようにコンテキストスイッチが頻繁に起こるシステムにおいて、コンテキストスイッチにかかる時間を一定とし、実時間性の時間粒度を非常に小さくすることを目的とする。

【0008】

【課題を解決するための手段】

本発明は、特に、

1. コンテキストを保持するための専用記憶装置（コンテキストキャッシュ）を持ち、
 2. レジスタよりビット幅の広い専用バスを用いて専用記憶装置（コンテキストキャッシュ）と中央演算装置（CPU）を接続する、
- ことにより、コンテキスト切り替えにかかるオーバーヘッドを削減する。

【0009】

本発明の第1の解決手段によると、
複数のコンテキストを切り替えるコンテキスト切り替え装置であって、
演算論理ユニット又はメモリアクセスユニットで実行すべきスレッドに関するコンテキストが記憶されたレジスタファイルと、
前記レジスタファイルに接続され、コンテキストをキャッシュするためのコンテキストキャッシュと、
前記レジスタファイルと前記コンテキストキャッシュを接続するコンテキスト

スイッチ専用バスと、

前記コンテキストキャッシュに記憶されているスレッドのコンテキストを識別するためのスレッド識別子（スレッドID）を記憶するスレッドIDテーブルを有し、演算論理ユニット及びメモリアクセスユニットと並列に接続され、前記コンテキストキャッシュと前記レジスタファイルとの間のコンテキストのデータ転送を制御するスレッド制御ユニットとを備え、

前記スレッド制御ユニットは、

コンテキストの切り替えが発生した場合、入力された切り替え命令と新たに入れ替えるスレッド識別子（スレッドID）に基付き前記スレッドIDテーブルを検索し、

新たに入れ替えるコンテキストが記憶されている前記コンテキストキャッシュのアドレスと、実行中のコンテキストが記憶されている前記レジスタファイルのレジスタ識別子を求め、

求めたアドレスに基付き前記コンテキストキャッシュをアクセスし、且つ、求めたレジスタ識別子に基付き前記レジスタファイルをアクセスし、前記レジスタファイル及び／又は前記コンテキストキャッシュのコンテキストを前記コンテキストスイッチ専用バスを介して入れ替え、退避又は復帰を実行する前記コンテキスト切り替え装置が提供される。

【0010】

本発明の第2の解決手段によると、

上述のようなコンテキスト切り替え装置と、

命令及びデータをそれぞれキャッシュする命令キャッシュ及びデータキャッシュと、

前記命令キャッシュから命令をフェッチ及びデコードする命令フェッチユニットと、

レジスタファイルに記憶された命令に従い各種演算を行い、演算結果を前記レジスタファイルに書き戻す演算論理ユニットと、

前記レジスタファイルからオペランドと命令が送られ、前記データキャッシュをアクセスし、ロード又はストアを実行するメモリアクセスユニットと、前記レジスタファイル、前記演算論理ユニット、前記メモリアクセスユニット及び前記スレッド制御ユニットを並列に接続する演算バスと、を備えた中央演算装置が提供される。

【0011】

本発明の第3の解決手段によると、

演算論理ユニット又はメモリアクセスユニットで実行すべきスレッドに関するコンテキストが記憶されたレジスタファイルと、

前記レジスタファイルに接続され、コンテキストをキャッシュするためのコンテキストキャッシュと、

前記レジスタファイルと前記コンテキストキャッシュを接続するコンテキストスイッチ専用バスと

を備えたコンテキスト切り替え装置を用い、複数のコンテキストを切り替えるコンテキスト切り替え方法であって、

コンテキストの切り替えが発生した場合、入力された切り替え命令と新たに入れ替えるスレッド識別子（スレッドID）に基付き、前記コンテキストキャッシュに記憶されているスレッドのコンテキストを識別するためのスレッド識別子（スレッドID）を記憶する前記スレッドIDテーブルを検索し、

新たに入れ替えるコンテキストが記憶されている前記コンテキストキャッシュのアドレスと、実行中のコンテキストが記憶されている前記レジスタファイルのレジスタ識別子を求め、

求めたアドレスに基付き前記コンテキストキャッシュをアクセスし、且つ、求めたレジスタ識別子に基付き前記レジスタファイルをアクセスし、前記レジスタファイル及び／又は前記コンテキストキャッシュのコンテキストを前記コンテキストスイッチ専用バスを介して入れ替え、退避又は復帰を実行する

前記コンテキスト切り替え方法

が提供される。

【0012】

さらに、本発明の第4の解決手段によると、

演算論理ユニット又はメモリアクセスユニットで実行すべきスレッドに関するコンテキストが記憶されたレジスタファイルと、

前記レジスタファイルに接続され、コンテキストをキャッシュするためのコンテキストキャッシュと、

前記レジスタファイルと前記コンテキストキャッシュを接続するコンテキストスイッチ専用バスと

を備えたコンテキスト切り替え装置を用い、コンピュータが複数のコンテキストを切り替えるコンテキスト切り替えプログラムを記録したコンピュータ読み取り可能な記録媒体であって、

コンテキストの切り替えが発生した場合、入力された切り替え命令と新たに入れ替えるスレッド識別子（スレッドID）に基付き、前記コンテキストキャッシュに格納されているスレッドのコンテキストを識別するためのスレッド識別子（スレッドID）を記憶する前記スレッドIDテーブルを検索するステップと、

新たに入れ替えるコンテキストが記憶されている前記コンテキストキャッシュのアドレスと、実行中のコンテキストが記憶されている前記レジスタファイルのレジスタ識別子を求めるステップと、

求めたアドレスに基付き前記コンテキストキャッシュをアクセスし、且つ、求めたレジスタ識別子に基付き前記レジスタファイルをアクセスし、前記レジスタファイル及び／又は前記コンテキストキャッシュのコンテキストを前記コンテキストスイッチ専用バスを介して入れ替え、退避又は復帰を実行するステップとをコンピュータに実行させるためのコンテキスト切り替えプログラム、及び、そのプログラムを記録したコンピュータ読み取り可能な記録媒体が提供される。

【0013】

【発明の実施の形態】

1. 関連技術

なお、本実施の形態は、一例として、並列分散リアルタイム制御用 R e s p o

nsive Multi-Threaded (RMT) Processor
の中央演算装置上でコンテキストキャッシュとして利用されている。そこで、まず、本実施の形態が関連するCPUを有するコンピュータについて説明する。

【0014】

図1に、一般的なCPUを含むコンピュータの構成図を示す。

このコンピュータは、CPU10、メモリ20、I/O30、バス40を有する。CPU10は、バス40を介して、メモリ20、I/O30等と接続されている。CPU10は、レジスタファイル1、命令フェッチユニット2、命令キャッシュ3、演算論理ユニット (arithmetic logic unit、ALU) 4、メモリアクセスユニット5、データキャッシュ6、バスインタフェースユニット7、演算バス11を備える。

【0015】

レジスタファイル1は、汎用レジスタ (GPR)、浮動小数点レジスタ (FPR)、プログラムカウンタ (PC)、ステータスレジスタ (SR) 等の各種レジスタを含む。レジスタファイル1は、現在実行中のコンテキストを記憶する。命令キャッシュ3及びデータキャッシュ6は、例えば、SRAM、フリップフロップ (FF) 等の素子が用いられ、アクセス、読み出し、書き込み等の処理速度が速いものの記憶容量が小さい。一方、CPU10外部のメモリ20は、SDRAM、DRAM等の素子が用いられ、アクセス、読み出し、書き込み等の処理速度がキャッシュより遅いものの記憶容量が大きい。

【0016】

ALU4は、局所性原理に従い、命令キャッシュ3、データキャッシュ6又はメモリ20から必要なデータ及び命令を利用する。ALU4は、命令キャッシュ3又はデータキャッシュ6に処理に必要な命令又はデータがあればそれを利用し、なければメモリ20にアクセスして必要な命令又はデータを取得する。また、バスインターフェースユニット7は、命令キャッシュ3、メモリアクセスユニット5、データキャッシュ6とCPU外部のメモリ20、I/O30等をバス40を介して接続し、CPU内部と外部の間でデータの入出力を行うユニットである。演算バス11は、レジスタファイル1、ALU4、メモリアクセスユニット5を並列

に接続する演算パイプライン等のバスである。

【0017】

命令フェッチユニット2は、命令キャッシュ3へアドレス (Address) を出力し、命令キャッシュ3から命令 (instruction) をフェッチし、デコードする。デコードされた命令に従い、ALU4がレジスタファイル1から必要なオペランドを読み出す。ALU4は、それに従い各種演算を行い、演算結果をレジスタファイル1に書き戻す。ロード (Load) 又はストア (Store) 等のメモリアクセス命令の場合、デコードされた命令に従い、メモリアクセスユニット5がオペランドを読み出す。

【0018】

メモリアクセスユニット5は、データキャッシュ6にアクセスしLoad又はStoreを行う。Store命令の場合、メモリアクセスユニット5は、アドレスとデータをデータキャッシュ6に送り、データをデータキャッシュ6に格納する。Load命令の場合、メモリアクセスユニット5は、アドレスをデータキャッシュ6に送りデータキャッシュ6からデータを読み出す。読み出したデータは、レジスタファイル1に書き戻される。このとき、データキャッシュ6に求めるデータがなければ、メモリ20からそれを読み出す。

【0019】

このような一般的な構成の場合、レジスタファイル1に記憶されたコンテキストの退避はStore命令を用いて行う。

【0020】

図2に、一般的なCPUでのコンテキストスイッチのフローチャートを示す。

OS等のソフトウェアはStore命令を発行し、コンテキストスイッチを開始する。Store命令が発行されると、レジスタファイル1からデータが読み出され、メモリアクセスユニット5に送られる。そして、メモリアクセスユニット5は、データを格納するアドレスを計算し、データキャッシュ6にアクセスする (S101)。もし、データキャッシュミスが発生した場合 (S103)、データキャッシュ6はメモリ20からキャッシュラインを読み込む (S105)。一方、データキャッシュミスが発生しなかった場合 (S103) は、ステップS

107へ進む。データキャッシュ6は、メモリアクセスユニット5から送られてきたデータを適当なアドレスで自身のキャッシュ内に格納する(S107)。OS等のソフトウェアは、このStore命令の処理を退避すべきレジスタの数だけ繰り返す。すなわち、全てのレジスタの内容の保存が完了していない場合は、ステップS101へ戻り処理を続け、全てのレジスタの内容の保存が完了した場合は、ステップS111へ進む(S109)。

【0021】

つぎに、OS等のソフトウェアは、新しく実行されるコンテキストの復帰を、Load命令を用いて行う。実行中のコンテキストの保存が完了すると、OS等のソフトウェアは、Load命令を発行する。Load命令が発行されると、メモリアクセスユニット5は、データを読み込むためのアドレスを計算し、データキャッシュ6をアクセスする(S111)。もし、データキャッシュミスが発生した場合(S113)、データキャッシュ6はメモリ20からキャッシュラインを読み込む(S115)。データキャッシュミスが発生しなかった場合(S113)は、ステップS117へ進む。メモリアクセスユニット5は、データキャッシュ6からデータが返ってくる(S117)と、それをレジスタファイル1に書き戻す。このLoad命令を復帰すべきレジスタの数だけ繰り返す。すなわち、全てのレジスタ内容の読み込みが完了した場合は、ステップS111へ戻り処理を続け、全てのレジスタ内容の読み込みが完了した場合は、コンテキストスイッチを終了する(S119)。

【0022】

2. コンテキスト切り替え装置を備えたCPU

図3に、コンテキストキャッシュを用いたCPUを含むコンピュータの構成図を示す。

このCPU100は、レジスタファイル1、命令フェッチユニット2、命令キャッシュ3、ALU4、メモリアクセスユニット5、データキャッシュ6、バスインターフェースユニット7、コンテキストキャッシュ8、スレッド制御ユニット9、演算バス11、コンテキスト専用バス12を備える。演算バス11は、レジスタファイル1、ALU4、メモリアクセスユニット5にさらにスレッド制御

ユニット 9 を並列に接続する演算パイプライン等のバスである。図 1 の CPU 10 の各部に対応して、同符号で示される各部の構成及び動作は、上述した通りである。

【0023】

コンテキストキャッシュ 8 は、SRAM、FF 等の素子が用いられ、アクセス、読み出し、書き込み等の処理速度が速い。コンテキストキャッシュ 8 は、レジスタファイル 1 にコンテキストスイッチ専用バス 12 により接続され、コンテキストをキャッシュするために用いられる。スレッド制御ユニット 8 は、コンテキストキャッシュ 8 を制御するためのユニットであり、ALU 4 やメモリアクセスユニット 5 と並列に接続されている。なお、スレッドとは、一般に、OS があるプロセス又はタスクを並列処理するため、プロセス又はタスクを分割した処理単位又は最小単位のことをいう。この際、プロセス又はタスクが分割されることなく、1 プロセス又は 1 タスクが 1 スレッドとなる場合もある。コンテキストスイッチが起こった場合には、現在実行されているスレッドのコンテキスト（汎用レジスタ、浮動小数点レジスタ、プログラムカウンタ、ステータスレジスタ等）を退避し、新しく実行されるスレッドのコンテキストを復帰する必要がある。本実施の形態によるコンテキストキャッシュ 8 を用いたコンテキストスイッチの場合、コンテキストの退避と復帰は、レジスタファイル 1 とコンテキストキャッシュ 8 のデータをコンテキストスイッチ専用バス 12 を経て入れ替えるスワップ (Swap) 命令を用いる。

【0024】

図 4 に、コンテキストキャッシュを用いた CPU でのコンテキストスイッチのフローチャートを示す。

OS 等のソフトウェアは、Swap 命令を発行し、コンテキストスイッチを開始する。Swap 命令は、入れ替えるスレッド ID と共にスレッド制御ユニット 9 に発行される (S201)。スレッド ID は、コンテキストキャッシュ 8 に格納されているスレッドを一意に識別するために用いる。スレッド制御ユニット 9 は、コンテキストスイッチ専用バス 12 を経て、レジスタファイル 1 からコンテキストキャッシュ 8 へデータを退避すると同時並列に、コンテキストキャッシュ

8からレジスタファイル1へ新しいスレッドのデータを送る。スレッド制御ユニット9は、送られてきたスレッドIDに基づいて自動的に必要な数だけレジスタファイル1内のデータとコンテキストキャッシュ8内のデータを入れ替える(S203)。このように、OS等のソフトウェアはSwap命令を発行するだけで、専用ハードウェアがコンテキストスイッチを開始し終了する。

【0025】

3. コンテキスト切り替え装置の詳細

図5に、コンテキスト切り替え装置の詳細構成図を示す。

レジスタファイル1は、汎用レジスタ111、浮動小数点レジスタ112、プログラムカウンタ113、ステータスレジスタ114を有する。コンテキストキャッシュ8は、コンテキストを格納するコンテキスト記憶領域8-1、8-2、・・・、8-nを所定のコンテキスト数有する。スレッド制御ユニット9は、オンチップのコンテキストキャッシュ8とレジスタファイル1を制御するためのコントローラである。スレッド制御ユニット9内部にはオンチップメモリに格納されているコンテキストを識別するためのスレッドIDを所定数保持するためのスレッドIDテーブル91を設ける。また、マルチスレッドプロセッサの場合は、レジスタファイル1が並列に複数存在することになる。

【0026】

図6に、Swap命令のフローチャートを示す。

コンテキストの切り替えは、例えばコンテキストスイッチハンドラー内でコンテキストスイッチ専用命令を用いることができる。OS等のソフトウェアが、Swap命令を発行すると、専用ハードウェアであるスレッド制御ユニット9が、そのSwap命令とスレッドIDを受けとる(S300)。スレッド制御ユニット9は、スレッドIDに基づいてスレッドIDテーブル91を検索し、コンテキストキャッシュ8をアクセスするため、入れ替えるべきスレッドのデータ(コンテキスト)が格納されているアドレスを計算し、また、レジスタファイル1をアクセスするため、レジスタIDを計算する(S301)。つぎに、全てのコンテキストのデータ(汎用レジスタ、ステータスレジスタなど)に対して、ステップS302からS304により繰返しループ処理が行われる。スレッド制御ユニッ

ト9は、計算したアドレスをもとにコンテキストキャッシュ8をアクセスし、新しく入れ替えるスレッドのデータ（コンテキスト）を読み出し、レジスタファイル1にそれを書き込む（S303）。それと同時又は並列に、スレッド制御ユニット9は、レジスタファイル1にアクセスし、今実行されているスレッドのデータ（コンテキスト）を読み出し、コンテキストキャッシュ8にそれを書き込む（S303）。このようにして、レジスタファイル1とコンテキストキャッシュ8は、それぞれのデータを入れ替える。全てのデータの入れ替えが終了していない場合（S304）、スレッド制御ユニット9は、コンテキストキャッシュ8をアクセスするアドレスとレジスタファイル1をアクセスするレジスタIDを1つずつ増やし、ステップS303に戻って処理を続ける（ステップS302及びS304による繰返しループ処理）。全てのデータの入れ替えが終了した場合（S304）、Swap命令を終了する。

【0027】

このように、スレッド制御ユニット9は、コンテキストキャッシュ8とレジスタファイル1のそれぞれのアドレスを連続的にインクリメントすることにより、必要な数だけデータを入れ替える。この際、従来のようにLoad及びStore命令を用いた場合には数百～千数百クロックサイクルかかっていたコンテキストスイッチを、本実施の形態では、レジスタファイル1とコンテキストキャッシュ8のコンテキストスイッチ専用バス12によるデータ転送幅をレジスタのビット幅に比べて非常に広くすることによって、1クロックサイクル～数クロックサイクル程度で実現する。具体的には、複数のレジスタをまとめて1つの巨大なレジスタとして、その巨大なレジスタ毎に前記レジスタIDを割り当てるようにする。この際、全てのレジスタをまとめて1つの巨大なレジスタとした場合、1クロックサイクルでコンテキストスイッチを実現できる。

【0028】

スレッド制御ユニット9は、コンテキストを入れ替えるSwap命令の他に、コンテキストを退避するバックアップ（Backup）命令、コンテキストを復帰するリストア（Restore）命令を処理する。Backup命令の場合は、コンテキストキャッシュ8からレジスタファイル1へのコンテキストのデータ

転送は行われず、レジスタファイル1からコンテキストキャッシュ8へのコンテキストのデータ転送のみが実行される。一方、Restore命令の場合は逆にレジスタファイル1からコンテキストキャッシュ8へのコンテキストのデータ転送は行われず、コンテキストキャッシュ8からレジスタファイル1へのコンテキストのデータ転送のみが実行される。

【0029】

つぎに、図7に、コンテキスト切り替え装置の実装に関する説明図を示す。

コンテキストキャッシュ8は、コンテキストをバックアップするオンチップのメモリであり、この例では2ポートを有し、CPUにオンチップで実装されている。この例では、コンテキストキャッシュ8は、書き込みポート82、読み出しポート83を備える。コンテキストキャッシュ8は、所定の複数コンテキスト分（例えば、32個等）の記憶領域を含むことができる。

【0030】

レジスタファイル1は、通常のリードポートとライトポートの他にコンテキスト切り替え用の専用ポート、即ちコンテキストスイッチ用読み出しポート17とコンテキストスイッチ用書き込みポート18を付加し、これらのポートにコンテキストを保持するための記憶装置を接続する。この例では、レジスタファイル1は、レジスタ読み出しポート15、レジスタ書き込みポート16、コンテキストスイッチ用読み出しポート17、コンテキストスイッチ用書き込みポート18を備える。レジスタ読み出しポート15は、レジスタファイル1からCPU内の装置へのレジスタの読み出しを、レジスタ書き込みポート16は、CPU内の装置からレジスタファイル1へのレジスタの書き込みを、コンテキストスイッチ用読み出しポート17は、レジスタファイル1からコンテキストファイル8へのレジスタの読み出しを、コンテキストスイッチ用書き込みポート18は、コンテキストファイル8からレジスタファイル1へのレジスタの書き込みを、それぞれ行うポートである。

【0031】

コンテキストの切り替えが発生した場合、OS等のソフトウェアはSwap命令を発行し、スレッド制御ユニットが、これらのポート17、18を通して中央

演算装置のコンテキストキャッシュ 8 内のデータをレジスタファイル 1 に保持し、新しいコンテキストをこのレジスタファイル 1 から取り出す。また、レジスタファイル 1 とコンテキストキャッシュ 8 を接続するコンテキスト専用バス 12-1、12-2 の幅をレジスタファイル 1 のビット幅より広くすることにより、1 度に入れ替えるデータの量を増やす。この例では、レジスタファイル 1 とオンチップメモリのコンテキストキャッシュ 8 とは、256 ビットのコンテキストスイッチ専用バス 12-1、12-2 で繋がれる。また、コンテキストキャッシュ 8 として 2 ポートのオンチップメモリを用いることにより読み込みと書き出しを同時に行うことができる。そのため 32 ビット、32 本の汎用レジスタは 4 クロックサイクルでコンテキストを入れ替えることができる。

なお、ビット数、記憶容量、ポート数等の各種パラメータは一例に過ぎず、適宜設定することができる。

【0032】

4. その他

本発明のコンテキスト切り替え方法又はコンテキスト切り替え装置・システムは、その各手順をコンピュータに実行させるためのコンテキスト切り替えプログラム、コンテキスト切り替えプログラムを記録したコンピュータ読み取り可能な記録媒体、コンテキスト切り替えプログラムを含みコンピュータの内部メモリにロード可能なプログラム製品、そのプログラムを含むサーバ等のコンピュータ、等により提供されることができる。

【0033】

【発明の効果】

本発明によると、以上のように、特に、リアルタイム OS などコンテキストの切り替えが頻繁に発生するアプリケーションにおいて、コンテキストの切り替え（コンテキストスイッチ）によるオーバーヘッドを大幅に削減することができる。また、本発明によると、例えば、コンテキストを切り替える度に、コンテキストの保存と読み込みのためのメモリアクセスが、1 サイクルから数サイクルで可能とするコンテキスト切り替え方法及び装置、中央演算装置、コンテキスト切り替えプログラムを記録したコンピュータ読み取り可能な記録媒体を提供すること

ができる。

【0034】

また、本発明によると、特に、実時間処理システムのようにコンテキストスイッチが頻繁に起こるシステムにおいて、コンテキストスイッチにかかる時間を一定とし、実時間性の時間粒度を非常に小さくすることができる。

【図面の簡単な説明】

【図1】

一般的なCPUを含むコンピュータの構成図。

【図2】

一般的なCPUでのコンテキストスイッチのフローチャート。

【図3】

コンテキストキャッシュを用いたCPUを含むコンピュータの構成図。

【図4】

コンテキストキャッシュを用いたCPUでのコンテキストスイッチのフローチャート。

【図5】

コンテキスト切り替え装置の詳細構成図。

【図6】

Swap命令のフローチャート。

【図7】

コンテキスト切り替え装置の実装に関する説明図。

【図8】

コンテキスト切り替え動作の説明図。

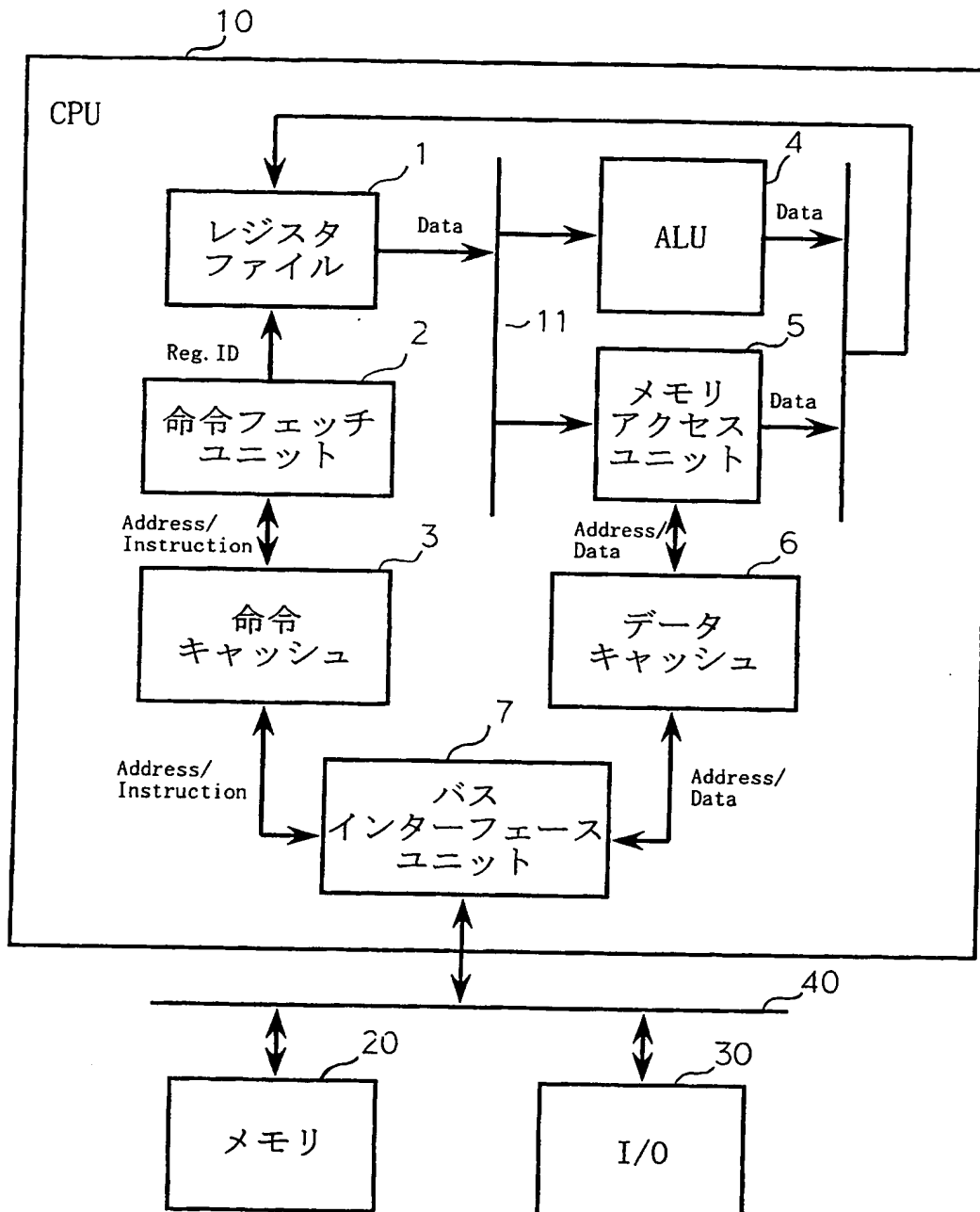
【符号の説明】

- 1 レジスタファイル
- 2 命令フェッチユニット
- 3 命令キャッシュ
- 4 ALU
- 5 メモリアクセスユニット

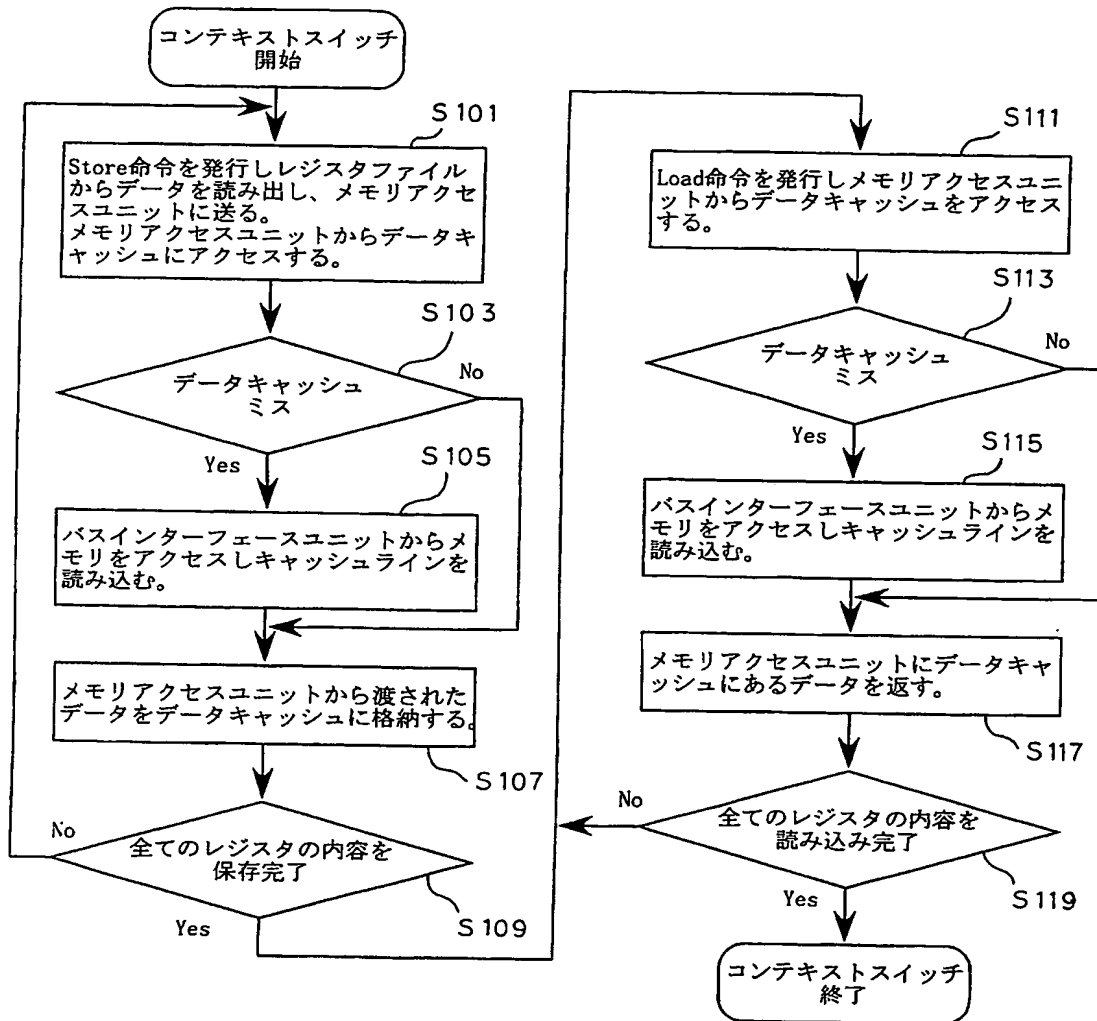
- 6 データキャッシュ
- 7 バスインタフェースユニット
- 8 コンテキストキャッシュ
- 9 スレッド制御ユニット
- 10 CPU
- 11 演算バス
- 12 コンテキスト専用バス
 - 12-1 コンテキスト専用バス
 - 12-2 コンテキスト専用バス
- 15 レジスタ読み出しポート
- 16 レジスタ書き込みポート
- 17 コンテキストスイッチ用読み出しポート
- 18 コンテキストスイッチ用書き込みポート
- 20 メモリ
- 30 I/O
- 40 バス
- 8-1~n コンテキストテーブル
- 91 スレッドIDテーブル
- 100 CPU
 - 111 汎用レジスタ
 - 112 浮動小数点レジスタ
 - 113 プログラムカウンタ
 - 114 ステータスレジスタ

【書類名】図面

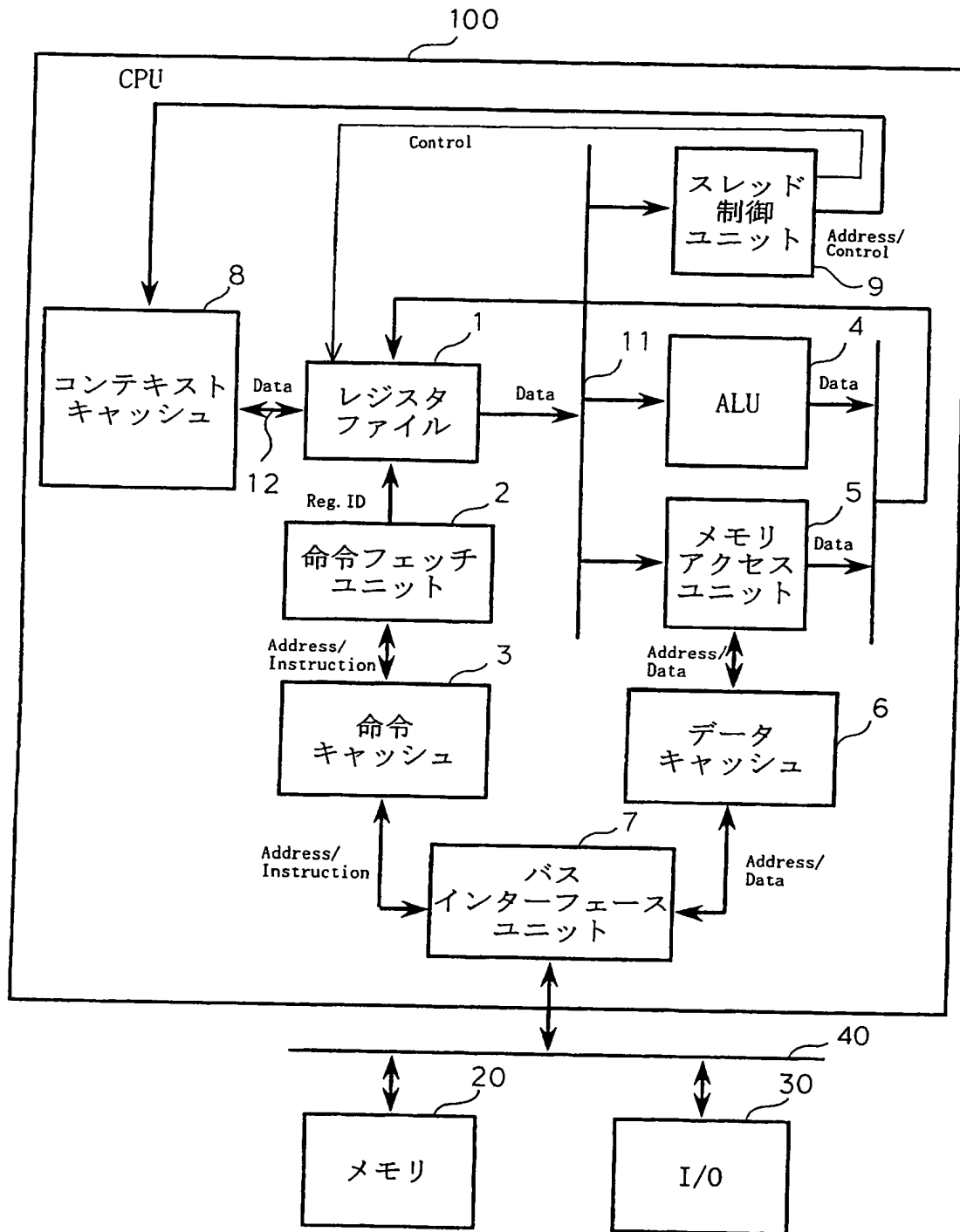
【図1】



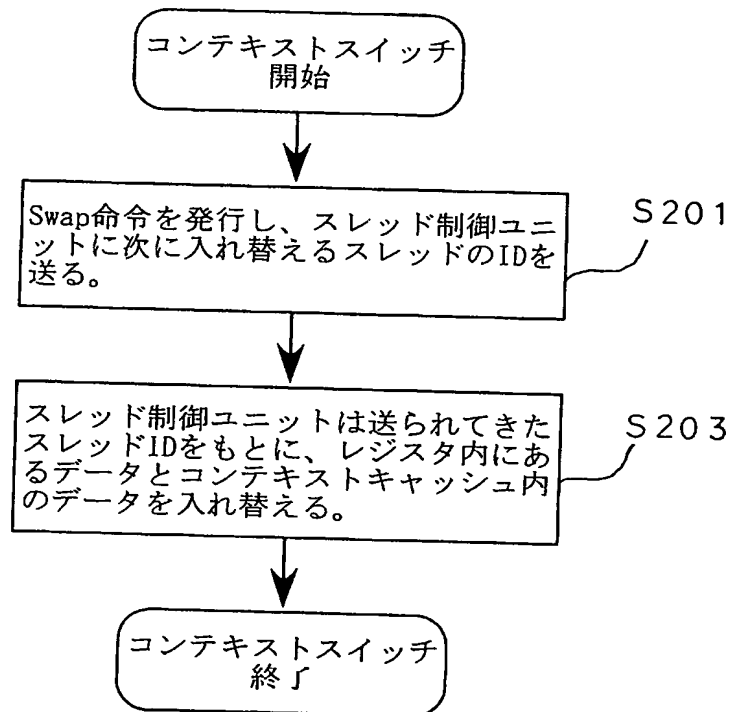
【図2】



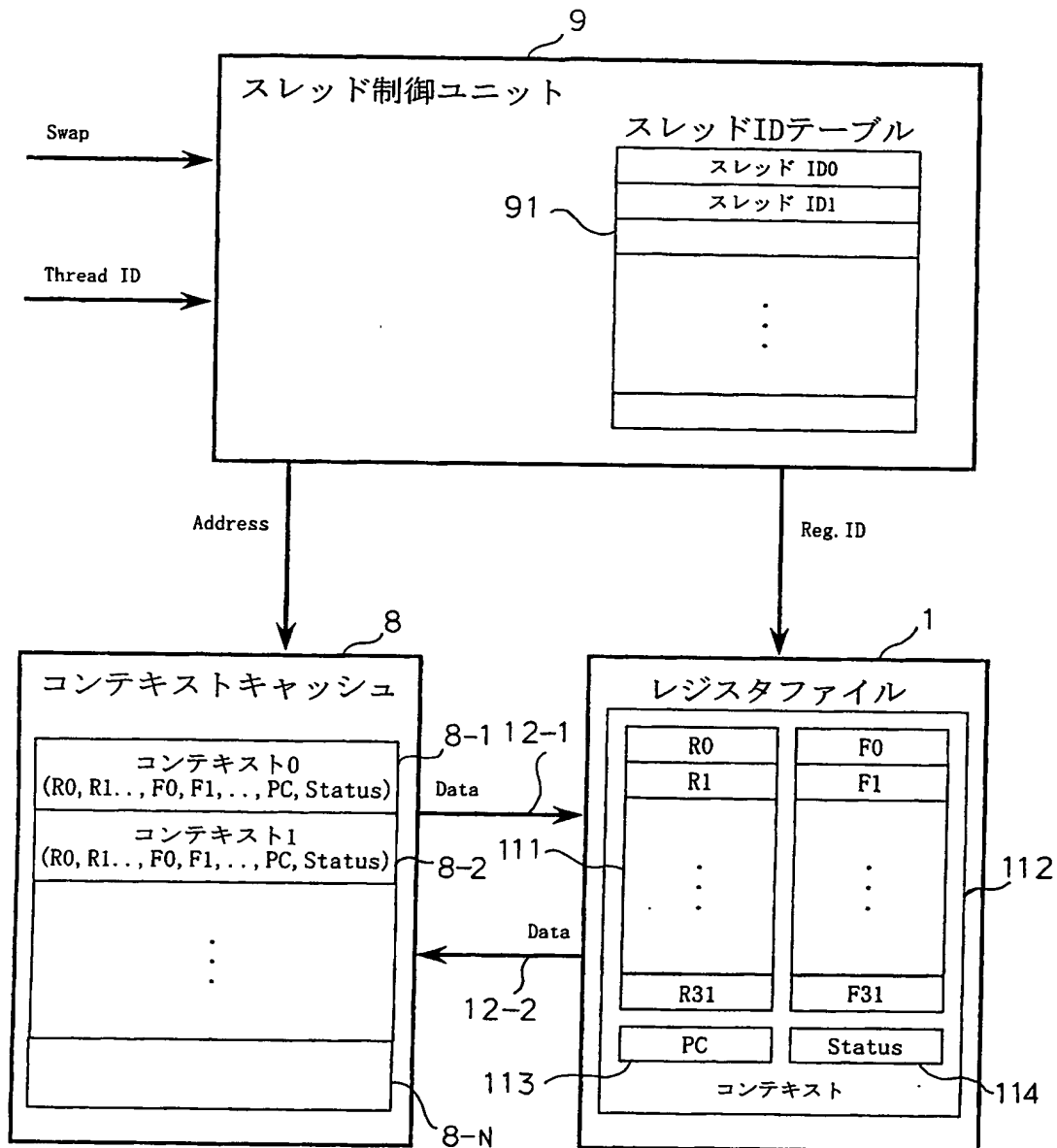
【図3】



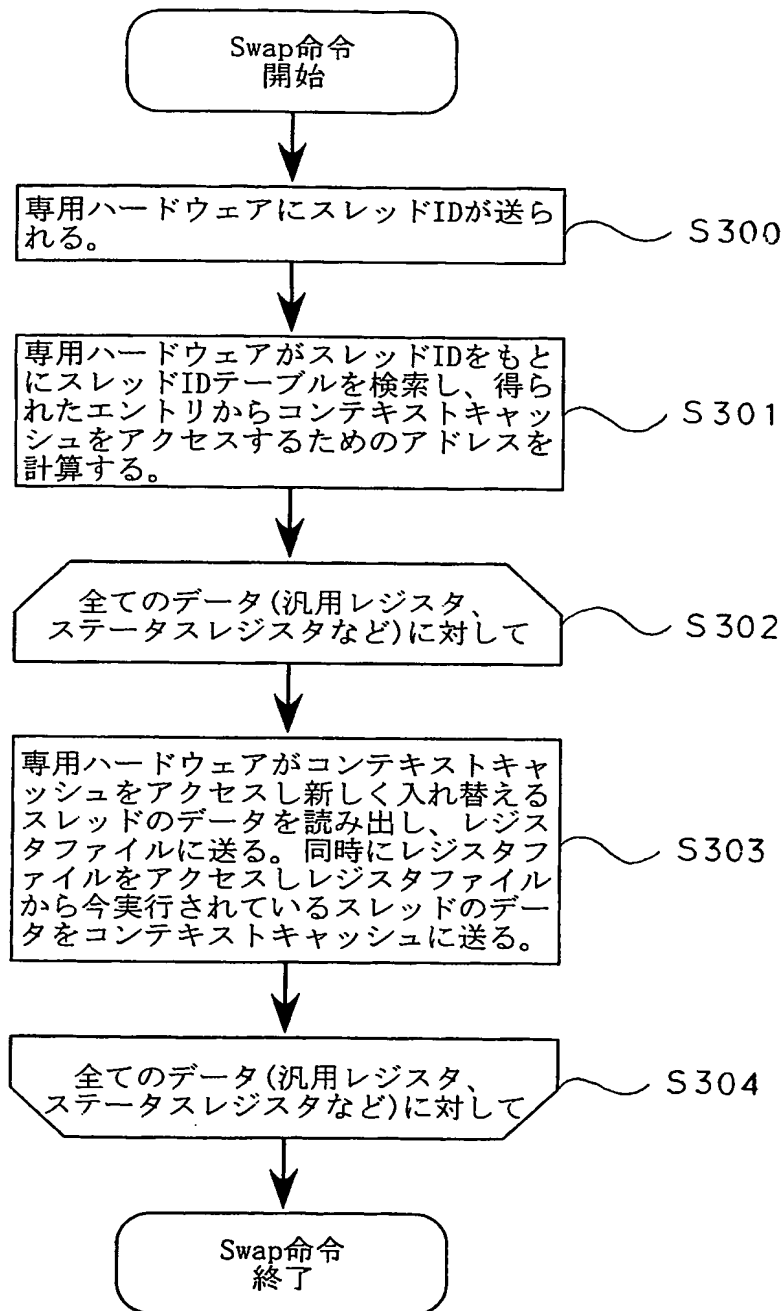
【図4】



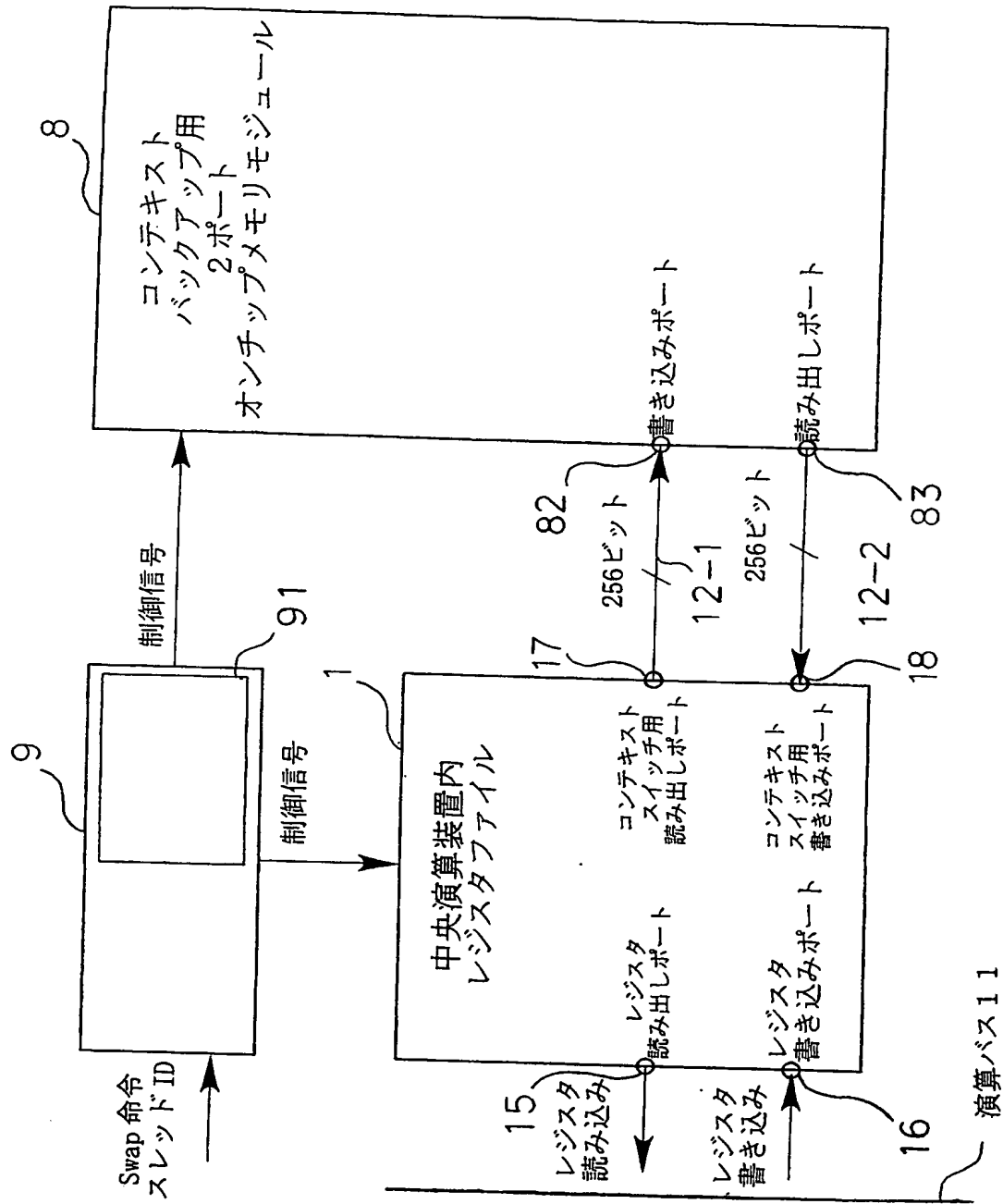
【図 5】



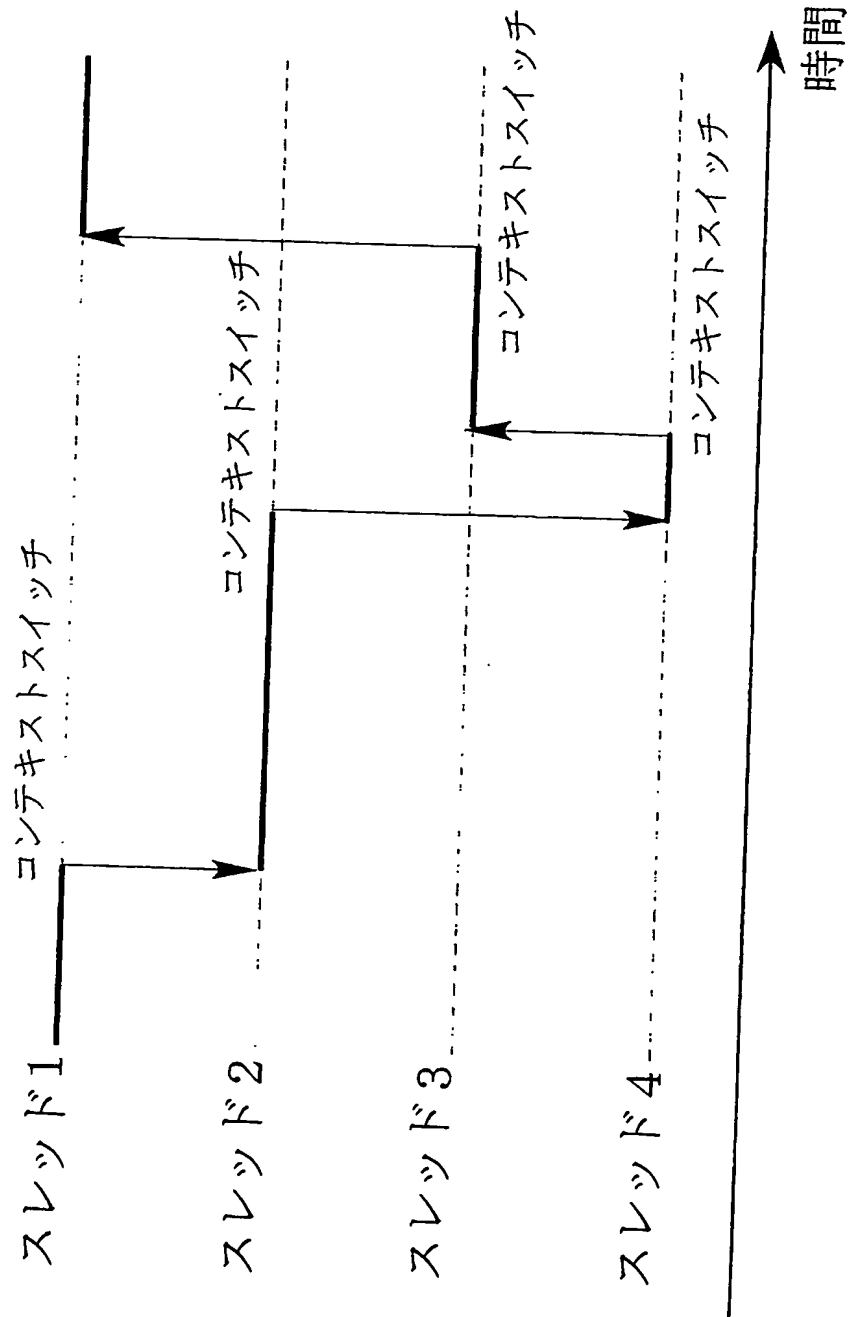
【図6】



【図 7】



【図8】



【書類名】 要約書

【要約】

【課題】 リアルタイムOSなどコンテキストの切り替えが頻繁に発生するアプリケーションにおいて、コンテキストの切り替えによるオーバーヘッドを大幅に削減する。

【解決手段】 OSは、Swap命令を発行し、コンテキストスイッチを開始する。Swap命令は、入れ替えるスレッドIDと共にスレッド制御ユニット9に発行される。スレッドIDは、コンテキストキャッシュ8に格納されているスレッドを一意に識別するために用いる。スレッド制御ユニット9は、コンテキスト専用バス12を経て、レジスタファイル1からコンテキストキャッシュ8へデータを退避すると同時に、コンテキストキャッシュ8からレジスタファイル1へ新しいスレッドのデータを送る。スレッド制御ユニット9は、送られてきたスレッドIDに基づいて自動的に必要な数だけレジスタファイル1内のデータとコンテキストキャッシュ8内のデータを入れ替える。

【選択図】 図3

【書類名】	出願人名義変更届（一般承継）
【提出日】	平成15年10月31日
【あて先】	特許庁長官 殿
【事件の表示】	
【出願番号】	特願2003- 3038
【承継人】	
【識別番号】	503360115
【住所又は居所】	埼玉県川口市本町四丁目1番8号
【氏名又は名称】	独立行政法人科学技術振興機構
【代表者】	沖村 憲樹
【連絡先】	〒102-8666 東京都千代田区四番町5-3 独立行政法人科学技術振興機構 知的財産戦略室 佐々木吉正 TEL 03-5214-8486 FAX 03-5214-8417
【提出物件の目録】	
【物件名】	権利の承継を証明する書面 1
【援用の表示】	平成15年10月31日付提出の特第許3469156号にかかる一般承継による移転登録申請書に添付のものを援用する。
【物件名】	登記簿謄本 1
【援用の表示】	平成15年10月31日付提出の特第許3469156号にかかる一般承継による移転登録申請書に添付のものを援用する。

特願 2003-003038

ページ： 1

出願人履歴情報

識別番号

[396020800]

1. 変更年月日
[変更理由]
住所
氏名

1998年 2月24日
名称変更
埼玉県川口市本町4丁目1番8号
科学技術振興事業団

出証番号 出証特 2003-3112562

特願 2003-003038

ページ： 2

出 願 人 履 歴 情 報

識別番号

[899000079]

1. 変更年月日

1999年 9月17日

[変更理由]

新規登録

住 所

東京都港区三田2丁目15番45号

氏 名

学校法人慶應義塾

出 願 人 履 歴 情 報

識別番号

[503360115]

1. 変更年月日

2003年10月 1日

[変更理由]

新規登録

住 所

埼玉県川口市本町4丁目1番8号

氏 名

独立行政法人 科学技術振興機構